Object-Oriented Programming Language

Lab 2

1. Create an integer variable called `num` and assign 10 to it. Create a reference variable called `refNum` which is a reference to `num` and display its value. Also create a pointer to `num` called `ptrNum` and display the value it's pointing to and the value itself (also known as the "address").

   The output of this program will be something like:

```
Value of refNum: 10
Value ptrNum is pointing to: 10
Value of ptrNum: 0x7ffd6a61f304
```

2. Swap the values of two integers using pointers. You can use the following code if you want to.

```
void swap(int* num1, int* num2) {
    // Please fill this blank

}

int main() {
    int a = 5;
    int b = 10;
    std::cout << "Before swap: a = " << a << ", b = " << b << std::endl;
    swap(&a, &b);
    std::cout << "After swap: a = " << a << ", b = " << b << std::endl;
    return 0;
}
```

The output of this program will be:

```
Before swap: a = 5, b = 10
After swap: a = 10, b = 5
```

3. Define a `struct` representing a rectangular with fields for its length and width. Declare a rectangle struct in the main function and take the value of its length and width from the user. Display the area and the perimeter of the rectangle.

   The output of this program will be something like:

```
Enter the length:5
Enter the width:3
Area: 15
Perimeter: 16
```

4. Write a program to find the maximum and minimum elements in a vector of integers. The elements are taken from the user. [Hint: You can use `while(cin >> input)` to check if the input is integer type (here `input` is an integer type variable). So once the user inputs a non-integer variable (ex. A char ‘q’), the while loop is over.]

   The output of this program will be something like:

```
Enter the elements: 2 4 4 5 1 q
Maximum element: 5
Minimum element: 1
```

5. Write a program to sort a vector of integers in ascending order (without using the built-in `sort` function). The number of elements and the elements are taken from the user.

   The output of this program will be something like:

```
Enter the elements: 2 4 4 5 1 q
Sorted vector: 1 2 4 4 5
```